**Proceedings of the 2007 IEEE Symposium on**
**Computational Intelligence and Data Mining (CIDM 2007)**

# Distributed Document Clustering Using Word-clusters[*]

Debzani Deb and Rafal A. Angryk
Department of Computer Science
Montana State University, Bozeman, MT 59717, USA

*Abstract*–**Document clustering has become an increasingly important task in analyzing huge numbers of documents distributed among various sites. The challenging aspect is to analyze this enormous number of extremely high dimensional distributed documents and to organize them in such a way that results in better search and knowledge extraction without introducing much extra cost and complexity. This paper presents a distributed document clustering approach called Distributed Information Bottleneck (DIB). DIB adopts a two stage agglomerative Information Bottleneck (aIB) algorithm to generate local clusters. At the first stage, the high-dimensional document vector is significantly reduced by finding word-clusters. These word-clusters are then used to obtain document-clusters in the second stage. DIB then extracts compact but informative local models from these document-clusters and transfers them to a central site. At the global site, the local models, that are likely to describe the same document set, are first combined. The resultant local models are then clustered by using the aIB algorithm to produce a hierarchical organization of all distributed documents. Our experimental results demonstrate the robustness, efficiency and effectiveness of DIB approach to cluster distributed documents.**

## I. Introduction

As the amount of text information available online in the form of electronic publications, digital libraries, web pages, etc. is increasing rapidly, the need to be able to automatically organize this type of information is gaining more and more importance. Currently existence of gigabytes or even terabytes of textual information is typical in different government agencies, academia, research organizations etc. Driven by the wide spread of the Internet, increase in problem complexity and global cooperation, more and more of such organizations are becoming decentralized. As a result, massive sets of documents are typically distributed among various sites managed by the same organization as they are published and downloaded. Companies, now more than ever, are realizing the need to make sense out of these immense repositories of distributed and unstructured textual data. The challenging aspect of this problem, however, is to analyze this enormous number of distributed documents and to cluster them within a reasonable amount of time that results in better search and knowledge extraction without introducing much cost and complexity.

Traditionally, clustering algorithms have centralized character; they require complete access to all the documents. One straightforward solution to the task of clustering distributed documents is to collect all available documents by crawlers and place them in a centralized repository before running a centralized clustering algorithm. However, this approach has two major disadvantages:

- Additional communication cost and latency caused by transmitting large numbers of massive text files to the central site. For massive sets of high dimensional distributed documents, these costs are usually significant and therefore the approach becomes time consuming and limited by network bandwidth available at the remote site.
- The extraordinary storage space and computational power requirement at the central site for large organizations (e.g. NASA). It requires a monstrous start-up cost on a central server while the distributed resources remain unutilized.

One of the most common reasons that data is distributed among multiple servers is that none of these servers is capable of storing whole datasets in the first place. Even if an organization decides to bear the above costs and takes the initiative to build such a centralized warehouse, soon the resources will be exhausted because of the increasing enormity of distributed documents, not to mention the cost of regular warehouse updates. Thus, the traditional centralized clustering algorithms are not appropriate to cluster massive sets of documents spread over multiple distributed environments. A distributed clustering technique has been recognized as a practical and scalable solution in such case [1,2,3]. By exploiting the computation and storage capabilities of distributed resources, a distributed clustering approach can balance the computational load among multiple sites and therefore time-critical applications, such as search and browse, are likely to benefit by implementing such a system while reducing the processing cost.

The distributed clustering process usually deploys the following steps [1,2,3]: 1) a centralized clustering algorithm is applied to each distributed data site concurrently to produce a *local model* for each site. 2) All *local models* are then aggregated in central server site to produce the *global model* of data distribution. *Local models* (also known as *local representatives* or *cluster prototypes*) are only a fraction or

[*] This work was supported in part by the NASA Grant Consortium under Award No. M166-05-Z3184.

summarization of data contained in a local cluster. To ensure minimal transmission cost and better scalability, a local model should describe a local cluster in a very compact form. At the same time, the model should also be informative enough to produce quality global clusters at the central site. There is a number of distributed clustering algorithms [1,2,3] and in step one of the above process presented, they generate cluster prototypes in terms of: cluster center and radius [1], Gaussian distribution [2], or the best representative points [3]. Most of these prototypes are compact and contain useful information. As a result, they are capable of producing quality clusters out of distributed repositories of large data.

However, besides being a very large data set, document corpus possesses another unique characteristic which is the very high dimensionality of the feature vector. This is the place where most of the above mentioned approaches fail to work effectively. The vocabulary for a document corpus can easily be thousands of words. Therefore, even with the cluster prototypes utilized in previously mentioned distributed clustering approaches, it is possible to exhaust the server site quickly with the transmission of high dimensional cluster prototypes. Depending on the number of distributed sites and the centralized clustering technique adopted at each local site, these transmission costs can be prohibitively high. To limit the transmission cost, some form of approximation of the cluster prototypes is needed as well.

To our knowledge, RACHET [1] is the only approach that takes initiative to approximate cluster prototypes to reduce the transmission cost in distributed settings. They used centroid and radius as a local model to represent a local cluster at the global site and further approximated high dimensional centroid vector by 6-tuple descriptive statistics. However, their approach of approximation can only be applied for a distance-based hierarchical distributed clustering approach. Recently, other dimensionality reduction techniques have been utilized to cluster document dataset such as Information bottleneck [4-6] Information-Theoretic Co-Clustering [7], Frequent Itemsets [8] etc. Experimental results [4-8] show that these approaches, when applied to text data, outperform commonly used distance-based approaches in terms of cluster quality. This research utilized one of the above mentioned methods *Agglomerative Information Bottleneck (aIB).* We have chosen aIB algorithm implemented in [4] to generate local clusters in step one of our distributed clustering process.

In this paper, we propose a distributed document clustering approach called Distributed Information Bottleneck (DIB). In our approach, local models are generated by a two stage aIB algorithm and then merged into a global model. At the first stage, the dimensionality of each document vector is reduced significantly by finding *word-clusters*. Those *word-clusters* are then used to obtain *document-clusters* in the second stage. Each local site may contain various distributions of documents, and to ensure the quality, every document need to be well described by the clustering solution chosen at that site. Because of both of the above factors, the necessary number of local clusters to be generated at each local site may vary significantly. DIB automatically determines a suitable number of local clusters at each site, consulting an evaluation graph that depicts the quality of clustering solutions at different numbers of clusters. There is no constraint or user controlled parameterization required to achieve this. Our proposed local models are compact but useful in providing the necessary information during merging at the global site. To combine the local models into a global model of all distributed documents, DIB first identifies the local models that describe the same document datasets and merges them. Then the resultant local models are clustered by using the aIB algorithm at the global site and a hierarchical organization of all distributed documents is obtained.

The rest of this paper is organized as follows: section II provides the necessary background for *Information Bottleneck* (IB) method and elaborates the reasons that motivated us to utilize this as local cluster generation approach in our study. Section III discusses the other distributed clustering approaches, section IV describes different aspects of DIB implementation, section V discusses the experiments and results, and section VI concludes the paper.

## II. Background

In document clustering, the similarity between two documents can be measured as the similarity between their word conditional distributions, which is defined as follows:

$$p(y|x) = \frac{n(y|x)}{\sum_{y \in Y} n(y|x)}$$

where, $n(y|x)$ denotes the number of occurrences of the word $y$ in the document $x$. Intuitively, documents with similar conditional word distributions are likely to belong to the same cluster. In their information bottleneck [5] approach, Tishby et al. proposed a novel technique to measure the distance between these distributions. The basic idea of their approach is: given the joint distribution of two variables *p(X,Y)*, one tries to find a compact representation of variable *X* so that the mutual information about variable *Y* is preserved as much as possible. The mutual information, *I(X;Y)*, between the random variables *X* and *Y* is given by

$$I(X;Y) = \sum_{x \in X, y \in Y} p(x)p\left(y|x\right)\log\frac{p\left(y|x\right)}{p(y)}$$

The authors argued that the above formula measures both the compactness of the representation and the preserved relevant information and thereby it can be formulated as a tradeoff between these quantities. The compactness is determined by *I(T;X)*, where *T* is a compressed representation of *X*, while the quality of the clusters, *T*, is measured by the fraction of the information they capture about *Y*, i.e. $I(T;Y)/I(X;Y)$. The authors found an exact optimal solution of this general problem that involves three distributions that characterize every cluster $t \in T$: the prior probability for this cluster, *p*(*t*), its membership probabilities,

$p(t|x)$, and its distribution over the relevance variable, $p(y|t)$. (Readers interested in detailed description of this process are referred to [5].) The IB principle determines the distortion between points $x$ and $t$ to be the Kullback-Leibler divergence [9] between the conditional distributions $p(y|x)$ and $p(y|t)$, defined as follows:

$$D_{KL}(p(y|x)\|p(y|t)) = \sum_y p(y|x)\log\frac{p(y|x)}{p(y|t)}$$

The formal solution is given by the following equations which must be solved together

$$p(t\mid x) = \frac{p(t)}{z(\beta,x)}\exp(-\beta D_{KL}(p(y|x)\|p(y|t))$$

$$p(y\mid t) = \frac{1}{p(t)}\sum_x p(t|x)p(x)p(y|x)$$

$$p(t) = \sum_x p(t|x)p(x)$$

where $z(\beta,x)$ is a normalization factor, and the single positive (Lagrange multiplier) parameter $\beta$ determines the tradeoff between compression and quality. Intuitively, in IB, the information contained in $X$ about $Y$ is compressed in a compact representation $T$, that is forced to represent the correlated part in $X$ with respect to $Y$.

Based on the Information Bottleneck method, agglomerative information bottleneck implemented in [4] starts with the trivial partitioning into $|X|$ clusters, where each cluster contains a single element of the document data set $X$. At each step, two clusters, among the set of $|X|$ clusters, are merged to a new cluster in a way that locally minimizes the loss of mutual information $I(T;Y)$. In [4], the problem of finding document clusters is resolved by a two stage algorithm. Based on the joint probability distribution of the set of documents and the set of words, the aIB algorithm first extracts *word-clusters* that capture most of the information about the documents. In the second stage, the original representation of the documents, the co-occurrence matrix of documents versus words, is replaced by a much more compact representation, i.e. the co-occurrence matrix of documents versus *word-clusters*. Using this new document representation, the same clustering procedure is applied again to obtain the desired *document-clusters*. Following are the reasons why we chose this particular method to generate local clusters in our study:

- *Reduced dimensionality*: Although the size of the vocabulary for a document corpus is very high dimensional, a single document often contains only a small fraction of words listed in the vocabulary. As a result, the initial co-occurrence matrix is usually sparse and highly dimensional. By employing the double-clustering method described above, it is possible to significantly reduce the noise of the original co-occurrence matrix. This approach results to a denser and more robust matrix based on *word-clusters*, which also reflects inherent structure of the document corpus better.
- *More accurate clusters*: Experimental results in various implementations based on the information bottleneck method show that the method outperforms other document clustering algorithms by a considerable margin [4-6].
- *Easy to browse and navigate structure*: The aIB algorithm produces hierarchical organization of documents, where each cluster is represented by a set of *word-clusters* that are highly correlated with the true topic of the enclosing documents. It is possible to extract sensible topics from those *word-clusters* and utilize them for effective browse and navigation.

## III. Related Works

Dhillon and Modha [10] developed a parallel implementation of the K-means clustering algorithm on distributed memory multiprocessors. A data set of size $n$ is divided into $P$ blocks of roughly equal size. During a K-means iteration, each distributed site updates the current $K$ centroids based on the local data and then broadcasts their centroids. After receiving all the centroids from other sites, a site can form the global centroids by averaging. The approach used in this study is straightforward to implement and comprehend. However, the implementation only generates flat clustering of data set and requires the user to supply the value of $K$ as a parameter. On the contrary, our goal is to find the hierarchical organization of data that does not require supplying number of clusters as an input.

Forman and Zhang [11] took an approach similar to the above [9], but extended it to K-harmonic means. Eisenhardt et al. [12] extended K-means with a 'probe and echo' mechanism for updating cluster centroids. Along with the problems associated with K-means clustering mentioned above, this approach generates a lot of message traffic in the network because of its peer to peer communication model.

Lazarevic and Obradovic [13] developed a distributed clustering algorithm to learn regression models from spatial datasets. However, the assumption made in their study is that each site should generate the same number of clusters, which seems not to be well-justified for real life problems.

Johnson and Kargupta [14] applied hierarchical clustering to distributed heterogeneous datasets. A chosen hierarchical algorithm is applied to generate local dendrogram based on the data available at each local site. These local dendrograms are then transmitted to a global site and merged to a global dendrogram using statistical bounds. Their algorithm only can be applied in case of vertically partitioned data (features are different in different distributed sites), which typically is not the case in document clustering task.

Samatova et al. [1] developed a method for merging hierarchical clusterings of homogeneously distributed data. Like in [14], their approach also produces local dendrogram at each local site and then later aggregates them to a global dendrogram. To reduce communication cost, instead of sending a complete description of each cluster, they only send an approximation of each cluster to the merger site. They generate descriptive statistics that approximately characterize each of the generated clusters in the local dendrogram.

Kriegel and Pfeifle [2] proposed a scalable and privacy-preserving distributed model-based clustering algorithm that uses Expectation Maximization (EM) to detect local models in terms of mixtures of Gaussian distributions. The authors then proposed an effective merging technique to merge these local Gaussian distributions to a meaningful global model. One interesting aspect of their study is the introduction of user level adjustment on issues like privacy preservation, transfer volume, and quality of the distributed clusters with respect to the centralized clusterings. However, their approach is expensive to apply in case of high dimensional dataset such as text documents. Initially they used $d \times d$ covariance matrix to represent each local cluster at the merger site. Later, they proposed another representation based on $d$-dimensional mean and variance matrix. Also, their *mutual support* technique used to combine local clusters into a global model requires evaluating integrals and therefore is expected to be time consuming.

Januaj and Kriegel [3] proposed a distributed version of DBSCAN. Few representatives are chosen at each local site based on their suitability criteria, which takes the density-based clustering technique into account. These representatives are then sent to a server site where they are clustered with an enhanced density-based clustering algorithm. In their system, there is a tradeoff between cluster quality and communication overhead, therefore performance is increased with the transmission of an increasing number of local representatives. Additionally, real data objects ($d$-dimensional) are sent to the server site, thereby increasing the transfer data volume and decreasing scalability.

## IV. DISTRIBUTED IB (DIB)

To build a distributed clustering algorithm, we assume that there are $S = \{S_1, S_2, \ldots, S_{|S|}\}$ distributed sites available in the system and $n$ document objects with $d$ dimensions (or words) are horizontally distributed among those sites i.e. each site has the same set of words but a different set of data points. As discussed in section I, a typical distributed clustering approach first analyses the local data independently and generates local clusters at each distributed site and in a subsequent step combines those local clusters to a global clustering of all documents. In this study, we consider a new approach to generate local *document-clusters*. Fig. 1 outlines the flow of operation.

Each site that contains the documents first generates *local word-clusters* based on the co-occurrence matrix representation of document versus words. As in [4], given a joint distribution of two variables $p(X,Y)$, where variables $X$ and $Y$ correspond to the set of documents and the set of words respectively, each word $y$ is represented by its conditional distribution over the set of documents, $p(x|y)$. The aIB algorithm is applied at each local site to obtain *local word-clusters* at that site. The *local word-clusters* generated at all local sites are then transmitted to the global site where they

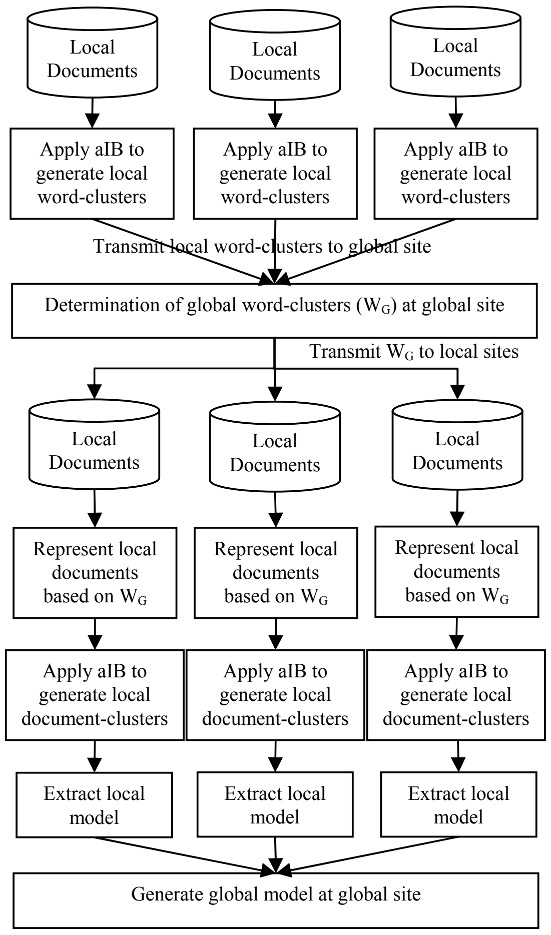
Local Documents
Local Documents
Local Documents
Apply aIB to generate local word-clusters
Apply aIB to generate local word-clusters
Apply aIB to generate local word-clusters
Transmit local word-clusters to global site
Determination of global word-clusters (W_G) at global site
Transmit W_G to local sites
Local Documents
Local Documents
Local Documents
Represent local documents based on W_G
Represent local documents based on W_G
Represent local documents based on W_G
Apply aIB to generate local document-clusters
Apply aIB to generate local document-clusters
Apply aIB to generate local document-clusters
Extract local model
Extract local model
Extract local model
Generate global model at global site

Fig. 1. Various Stages of DIB

are merged to *global word-clusters, $W_G$*. The objective is to generate *word-clusters* that co-occur frequently across the local sites. Once generated, the *global word-clusters* are then transmitted back to the local sites.

At each local site, the *global word-clusters* are used to replace the original representation of documents in terms of words. Instead of representing a document by its word conditional distributions, we now represent it by its conditional distributions of the elements of the set $W_G$. Using the compact representation of documents based on *global word-clusters*, we apply the aIB algorithm to extract the desired *document-clusters* at each local site. Once generated, local models are then extracted from these *local document-clusters* and sent to the global site for merging. Important aspects of the above process are detailed in the following sub sections.

### *A. Generation of Global Word-clusters*

*Global word-clusters* can be defined as a set of clusters $W_G = \{W_{G_1}, W_{G_2}, \ldots W_{G_m}\}$, where each cluster $W_{G_i}$ is a subset of the set of words $Y$, extracted in such way that each element of $Y$ is covered by, at most, one $W_{G_i}$. The proposed algorithm that generates *global word-clusters* is shown in Fig. 2. It takes as input the set of all *local word-clusters* generated at all local sites and, by employing a greedy strategy, finds the *global word-clusters*. Our algorithm favors those sets of words that occur together in most of the local sites. In the case of distributed document clustering, it is very important to generate representations of local *document-clusters* with reduced dimensionality and by utilizing *global word-clusters* technique we achieve that goal. The dimensionality of each document vector is now reduced to the number of elements in the *global word-cluster* set $W_G$ instead of the size of the initial set of words.

Steps (a) and (d) of the algorithm execute for $|W_L|$ times, where $W_L$ is the set of *local word-clusters* generated in all local sites. Step (a.i) involves comparing an element of set $W_L$ against all other elements in the set. Therefore the complexity of step (a) is bounded by $O(|W_L|^2)$. Step (b) of the algorithm involves sorting the elements of $W_L$, and therefore the complexity of this step is bounded by $O(|W_L| log|W_L|)$. The highest value of $W_L$ can be $d.|S|$, and it may occur when each of the $d$ words form a separate cluster in all $S$ sites. However, in real life, the number of *word-clusters* generated at a local site is anticipated to be far less than $d$ and therefore the overall complexity of the algorithm should not be too high.

### B. Generation of Local Model

Our proposed local model represents each *document-cluster* $C$ by its centroid vector $\vec{c}$ and covering radius $R_c$, indicating the area represented by the cluster $C$, the same measures as in Rachet [1]. Cluster centroid $\vec{c} = (f_{c1}, f_{c2}, \ldots f_{c|W_G|})$ is the mean vector of all the document vectors in cluster $C$. More formally,

$$\vec{c} = \frac{1}{n_c}\sum_{i=1}^{n_c}\overrightarrow{p_i} \qquad f_{cj} = \frac{1}{n_c}\sum_{i=1}^{n_c} p_{ij}, j = 1,|W_G|$$

where, $n_c$ is the number of elements in cluster C and $p_{ij}$ is the $j$-th component of document vector $\overrightarrow{p_i}$. The radius $R_c$ of a cluster $C$ is defined as the average squared Euclidean distance of a point from the centroid of the cluster:

$$R_c = \sqrt{\frac{\sum_{i=1}^{n_c}\sum_{j=1}^{|W_G|}(p_{ij} - f_{cj})^2}{n_c}}$$

The centroid and radius representations of a cluster $C$ are considered to be a good approximation of all documents residing in $C$ [1,15], as the first represents the best representative and the second indicates the area covered by $C$.

### C. Generation of Global Model

Once extracted, the local models are transferred to the global site, where they are merged in order to reconstruct the global model. In the distributed clustering, the objective is to combine the local models into a global model in such a way so that the clusters generated from this global model (referred as *global clusters*) resemble the clusters generated from a

```
Input:  Set of unique words: U
        Set of local word-clusters generated in all local sites: W_L
Algorithm:
a. For each word-cluster C in W_L
       i.   Calculate the weight of C as number of times C
            appears as a whole in W_L or a subset of another
            cluster of W_L.
       ii.  Add C along with its weight to another set G.
b. Sort G according to the weights so that most weighted cluster
   appears first. When there are two or more clusters with same
   weight, place the larger one first.
c. Global word-cluster set, $W_G = \phi$.
d. While $U \neq \phi$
       i.   Select cluster T with largest weight in G, such
            that $(W_G \cap T) = \phi$
       ii.  $U = U -$ (words in T)
       iii. $G = G - \{T\}$
       iv.  $W_G = W_G \cup \{T\}$
Output: $W_G$ is the global word-clusters
```

Fig. 2. Global word-clusters generation

centralized approach (referred as *centralized clusters*). The challenging part is to determine which of these local *document-clusters* described by local models are likely to be a part of the same global cluster, and therefore should merge, and which entirely describes a global cluster and should be added to the set of generated global clusters.

To find out the global clusterings of all documents in the best possible way, we need a measure that determines the degree to which two local clusters, $C_1$ and $C_2$, represented by local models $(\vec{c}_1, R_{c_1})$ and $(\vec{c}_2, R_{c_2})$, intersect with each other. In this study, we used the Euclidian distance between centroid vectors $d\ (\overrightarrow{c_1}, \overrightarrow{c_2})$ as such a measure. Having the measure, we identified three cases that can occur while measuring the intersection between two local clusters $C_1$ and $C_2$:

1. $d\ (\overrightarrow{c_1}, \overrightarrow{c_2}) \geq R_{c_1} + R_{c_2}$, i.e. $C_1$ and $C_2$ do not intersect and each describes a well separated document data set.
2. $(d\ (\overrightarrow{c_1}, \overrightarrow{c_2}) \leq R_{c_1})\ or (d\ (\overrightarrow{c_1}, \overrightarrow{c_2}) \leq R_{c_2})$ i.e. $C_1$ is contained completely inside $C_2$ or $C_2$ is contained completely inside $C_1$. Obviously, when $C_1$ encloses $C_2$ completely, $C_2$ becomes redundant as $C_1$ models the same document data set from which cluster $C_2$ is developed.
3. $((d\ (\overrightarrow{c_1}, \overrightarrow{c_2}) < R_{c_1} + R_{c_2})\ and\ ((d\ (\overrightarrow{c_1}, \overrightarrow{c_2}) > R_{c_1})\ or$ $(d\ (\overrightarrow{c_1}, \overrightarrow{c_2}) > R_{c_2})))$, i.e. $C_1$ and $C_2$ overlaps.

As the first step of our global merging approach, the Euclidian distances between each pair of local *document-clusters* are calculated. After that, the pairs that follow case 2 are identified and the enclosed cluster is merged into the enclosing one. By doing this, we eliminate the possibility of having two clusters in the local *document-clusters* set that describe exactly the same data set. The remaining set of cluster centroids at the global site is then clustered by applying the aIB algorithm to build the global model of all distributed documents.

### *D. Determining Number of Local Clusters*

Another challenging aspect of any distributed clustering approach is automatically determining a suitable number of local clusters to be generated at each site based on the local data. As we can not make any assumption about the nature of the distribution of documents among the distributed sites, a variety of distributions may occur across the local sites. For example, all documents belonging to a particular global cluster may exist on a single local site, they may evenly or unevenly be distributed among all the nodes, or they may be distributed over only a few sites. Therefore, it is possible that the necessary number of *word-clusters* and *document-clusters* to be generated at each local site might strongly vary.

We also want to make sure that enough clusters are generated at each local site so that each word/document is well described by a corresponding cluster and the overall quality of the clustering is preserved. Most distributed approaches either explicitly provide the number of clusters to return [11-13], or they provide some other parameters that implicitly control the number of clusters to return, such as parameters that controls privacy level and transfer volume, utilized in [2]. It is important to note that such parameterization either requires detailed pre-existing knowledge of the data and its distribution, or requires multiple, time consuming and error prone experiments with different parameter values.

In this study, we utilized a methodology that does not require any background knowledge. It uses only limited number of well defined experiments to determine a reasonable number of clusters to be returned from a local site. Our approach makes use of an evaluation graph where the *x*-axis describes the number of clusters and the *y*-axis signifies the quality of the clustering with *x* clusters. As mentioned in section 2, the quality of an aIB generated clustering solution *T* is measured by $I(T;Y)/I(X;Y)$. The graph $|T|$ versus $I(T;Y)/I(X;Y)$ is a smooth, monotonically increasing graph, where the quality is maximized when $|T| = |X|$. Once we have this evaluation graph for different number of clusters, the knee, or the point of maximum curvature of this graph, is used as the number of clusters to return. Intuitively, the knee identifies the point at which further increase in quality requires huge increase in cost, i.e. the number of clusters. This approach is utilized in various centralized clustering approaches [16 and references thereafter]. In our case, the knee is measured as the largest ratio differences between two points of the curve. Therefore, in this study, at each local site the number of both *word-clusters* and *documents-clusters* are automatically determined based on the data available at that site and the evaluation graph.

## V. Experimental results

We implemented the DIB approach in Java and ran the experiments on 3 machines, 2 of them featuring a 3 GHz Pentium processor and 1 GB physical memory and the third contains a 2.8GHz processor with, 512 MB RAM.

### *A. Datasets*

Our main concern is to evaluate how well the clusters generated by our DIB approach resemble the clusters generated by a centralized approach and how stable the solution is in the context of various distributions of data (specially when many of the clusters have very similar topics). Therefore, we decided to evaluate our method on several, deliberately chosen subsets based on a standard, labeled corpus 20Newsgroup dataset [17]. This corpus contains about 20,000 articles evenly distributed among 20 UseNet discussion groups, some of which are of very similar topics. Having the knowledge about the correct labels of the documents allows us to choose different datasets of very similar topics and to enforce various distributions of data among distributed sites.

To preprocess and index corpus data, we used *Bow* [18], which is a toolkit for statistical language modeling. During the preprocessing, the following steps are performed by Bow:

1. File headers are removed from each document.
2. Documents containing *html* are skipped for simplicity.
3. Stop words (i.e., topic-neutral words such as articles (a, the), prepositions, conjunctions, etc.) are removed.
4. Stemming (i.e., grouping words that share the same morphological root) is performed. Thus all the words sharing the same stem (For example, “compute”, “computing” and “computer”) are considered to be the same word.

After preprocessing, the document corpus contains a vocabulary of 99,000 unique words. We further adopted a standard feature selection procedure, *Information Gain*, and reduced the vocabulary size to a more manageable size of 2000 words. More specifically, we sorted all words based on their contribution to the mutual information about the documents and then selected the most influential 2000. From this corpus, we generated two different medium scale data sets, where groups talk about very similar topics. Detailed characterization of the data set is presented in Table I.

To simulate distributed datasets, we used various portions of the above mentioned two corpuses. Specifically, we divided the large corpuses into *S* parts, where *S* is the number of distributed nodes, and then transferred them to the remote sites. Each Newsgroup message has been given a unique identification number so that it can be identified unambiguously across the distributed environment. To thoroughly test the robustness of our approach, we deliberately created different data distributions that simulate various real life scenarios. For example, when Corpus 2 is distributed among 3 nodes, we assigned all the documents of group *sci.crypt* and *talk.politics.guns* to machine #1 and #2 respectively. Documents of group *sci.electronics* and *talk.politics.mideast* were evenly distributed among 3 sites, documents of group *sci.med* were unevenly distributed among machine #1 and #3, documents of group *sci.space* were unevenly distributed among machine #1 and #3, and documents of *talk.politics.misc* are unevenly distributed among all three machines. Corpus 1 is on the other hand

TABLE I
DATASETS UTILIZED

| Corpuses | Newsgroup topics | # of docs | Total # |
|---|---|---|---|
| Corpus 1 | Talk.politics.guns | 910 | 2625 |
| | Talk.politics.mideast | 940 | |
| | Talk.politics.misc | 775 | |
| Corpus 2 | sci.crypt | 988 | 6284 |
| | sci.electronics | 961 | |
| | sci.med | 749 | |
| | sci.space | 978 | |
| | Talk.politics.guns | 904 | |
| | Talk.politics.mideast | 936 | |
| | Talk.politics.misc | 768 | |

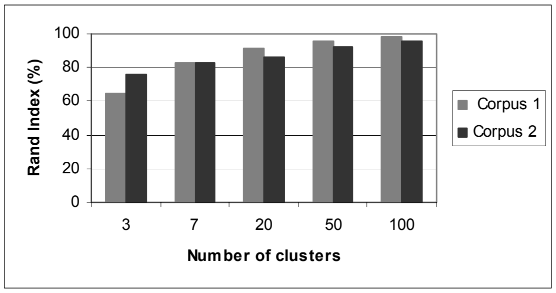

Fig.3. Cluster quality

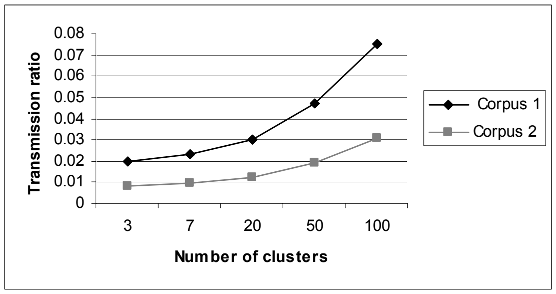

Fig. 4. Transmission ratio

equally distributed among the available nodes i.e. each machine receives equal portion of all three groups of documents of Corpus 1.

## B. Evaluation Method

We evaluated the strength of the DIB approach by comparing its results with the results of the centralized aIB algorithm by using *Rand Index* [19]. Computation of a Rand Index involves pairs of points that were assigned to the same and to the different clusters in each of two partitions. In our case, those two partitions are, $C = \{C_1, . . . ,C_m\}$, the clusterings of the whole data set in centralized settings, and $D = \{D_1, . . . ,D_s\}$, the clustering results of the distributed approach over the same data set. A pair of points ($x_u$, $x_v$) from the data set is referred using the following terms:

- *SS*: if both points belong to the same cluster of *C* and *D*.
- *SD*: if points belong to the same cluster of *C* and to different clusters of *D*.
- *DS*: if points belong to different clusters of *C* and to the same cluster of *D*.
- *DD*: if both points belong to different clusters of *C* and to different clusters of *D*.

Assuming now that *a, b*, *c* and *d* are the number of *SS*, *SD*, *DS* and *DD* pairs respectively, then $a+b+c+d = M$ which is the maximum number of all pairs in the data set. Now, the Rand Index is used to measure the degree of similarity between *C* and *D* as follows:

$$R = (a + d)/M$$

The above index takes values between 0 and 1, and are maximized when $m = s$. To evaluate the agreement between our algorithm and the centralized one, we performed both the centralized and distributed clustering with the predetermined equal number of clusters.

## C. Results and Discussion

Fig. 3 shows the quality of our approach compared to the centralized clustering in terms of Rand Index with respect to a different number of clusters. The figure demonstrates that the results of DIB approach highly resemble the results of the centralized approach. Corpus 1 results in better quality clusters in most cases. In Corpus 1, as the documents are evenly distributed among all sites, it becomes easier to detect clusters correctly and the results validate our expectations. The results of Corpus 2 show that DIB works well even in the presence of various distributions we enforced during building distributed corpuses. Near optimal rand index also indicates that our approach of reducing dimensions in terms of *global word-clusters* does not introduce any inefficiency in terms of clustering quality. Moreover, the resultant matrix, based on *word-clusters,* reduced the noise of the original co-occurrence matrix and thereby provides better quality *document-clusters*.

Fig. 4 shows the ratio of transmission cost (in bytes) of DIB approach compared to the transmission of all data from the local sites to a global site in centralized approach. We can see from Fig. 4 that, in terms of transfer cost, DIB is far more superior than the centralized approach. Based on the number of clusters and the corpuses utilized in this study, the centralized approach transfers 12 to 100 times more data than our DIB approach. In DIB, data is transferred between local sites and the global site on three occasions (Fig. 1) such as: 1) *local word-clusters* are transmitted to global site, 2) *global word-cluster* is transmitted back from global site to local sites, and 3) local models for *document-clusters* are sent to global site. Both local *word-clusters* and *global word-cluster* are a partitioning of all 2000 words among the different groups, therefore during cases 1) and 2), the same amount of data is transferred between each local site and the global site for both of the corpuses, irrespective of the number of clusters these 2000 word form optimally at each local site. Case 3, however, can generate varying transmissions due to the transfer of a different number of local models representing document-clusters generated at each site. Because of these reasons, transmission ratio in case of Corpus 2 is far lower than the transmission ratio of Corpus 1 which is approximately one third of Corpus 2 in terms of size. Therefore for bigger corpuses, the transmission ratio is expected to be lower in DIB.

## VI. CONCLUSIONS AND FUTURE WORKS

In this paper, we proposed a distributed document clustering approach called DIB. We applied a two stage agglomerative information bottleneck algorithm to generate local clusters. At the first stage, the high-dimensional document vector is reduced significantly by finding *word-clusters*. These *word-clusters* are then used to obtain *document-clusters* in the second stage. DIB then extracts compact but interpretable local models from these *document-clusters* generated at each site. We then proposed an efficient merging technique that combines these local models to a global model of all documents. Another significant contribution is the automatic determination of the number of local clusters to be generated at each site that does not require any parameterization or time consuming and error prone experimentation. Experimental results demonstrate that the agreement between DIB and the centralized approach is very high, which signifies the efficiency and robustness of our approach in the presence of various document distributions. DIB reduces the transmission cost dramatically and is expected to perform better in case of bigger corpuses. Our experiments also demonstrated that our approach of reducing dimension in terms of *global word-clusters* does not introduce any inefficiency in terms of clustering quality. Moreover, the resultant matrix, based on *word-clusters,* reduced the noise of the original co-occurrence matrix and thereby provides better quality *document-clusters*.

In conclusion, our experimental results, achieved on medium scale real-world datasets, with overlapping clusters and under various distribution scenarios demonstrate that DIB provides a comparable quality solution to the distributed document clustering problem, while minimizing the transfer cost. The reason for not using a larger dataset is that at this stage our focus is to justify different techniques used in DIB approach to create a comparable quality solution without increasing time, space and communication cost. The next step is to study the effectiveness of DIB in dealing with very large real datasets. We also plan to use larger number of distributed nodes. The experiments will be carried out to evaluate the scalability of our approach in case of both the number of data points and the number of data sites. In future, we also hope to extract sensible topics that are highly correlated with the true topic of the resultant *document-clusters* and then organize them for effective browsing and navigation.

## ACKNOWLEDGEMENT

The authors would like to thank the Montana NASA EPSCoR Grant Consortium for supporting this research.

## REFERENCES

[1]. N. F. Samatova, G. Ostrouchov, A. Geist, and A. Melechko, “RACHET: an efficient cover-based merging of clustering hierarchies from distributed datasets”, *Distributed and Parallel Databases*, vol. *11*(2), pp. 157-180, 2002.

[2]. H.-P. Kriegel and M. Pfeifle, “Effective and efficient distributed model-based clustering”, *Proceedings of the 5th international conference on Data mining (ICDM'05), pp. 285-265, 2005.*

[3]. E. Januzaj, H.-P. Kriegel and M. Pfeifle, “Scalable density-based distributed clustering”, *Proc. of 8th European Conf. on Principles and Practice of Knowledge Discovery in Databases (PKDD'04)*, Pisa, Italy, 2004, *In Lectures Notes in Computer Science*, Springer, vol. 3202, pp. 231-244, 2004.

[4]. N Slonim and N Tishby, “Document clustering using word clusters via the information bottleneck method”, *In the 23rd Annual International ACM SIGIR Conf. on Research and Development in Information Retrieval, 2000.*

[5]. N. Tishby, F.C. Pereira and W. Bialek, “The information bottlencek method”, *In Proc. of the 37-th Allerton Conference on Communication and Computation*, 1999.

[6]. N. Slonim, N. Friedman, and N. Tishby, “Unsupervised document classification using sequential information maximization. *In Proc. of SIGIR*, pp. 129-136, 2002.

[7]. S. Dhillon, S. Mallela, and D. S. Modha, “Information-theoretic co-clustering”, *In Proceedings of The Ninth ACM SIGKDD International Conference,* pp. 89-98, 2003.

[8]. B. C. M. Fung, K. Wang, and M. Ester, “Hierarchical document clustering using frequent itemsets”, *Proceedings of the SIAM International Conference on Data Mining*, 2003.

[9]. T. M. Cover and J. A. Thomas, *Elements of Information Theory,* John Wiley & Sons, New York, 1991.

[10]. S. Dhillon, and D. S. Modha, “A data-clustering algorithm on distributed memory multiprocessors, *Proceedings of International Conference of Large-Scale Parallel Data Mining*, pp.245-260, 1999.

[11]. G. Forman, and B. Zhang, “Distributed data clustering can be efficient and exact”, *SIGKDD Explorations*, vol. *2*(2), pp.34-38, 2000.

[12]. M. Eisenhardt, W. Muller, and A. Henrich, “Classifying documents by distributed P2P clustering”, *In Proceedings of Informatik 2003*, *GI Lecture Notes in Informatics*, Frankfurt, Germany, September 2003.

[13]. D. Lazarevic Pokrajac, and Z. Obradovic, “Distributed clustering and local regression for knowledge discovery in multiple spatial databases”, *In Proceedings of the 8th European Symposium on Artificial Neural Networks*, pp. 129-134, 2000.

[14]. E. Johnson and H. Kargupta, “Collective, hierarchical clustering from distributed, heterogeneous data”, *In Lecture Notes in Computer Science*, vol. *1759*, Springer-Verlag, pp. 221-244, 1999.

[15]. D. Deb, M. M. Fuad and R. A. Angryk, “Distributed hierarchical document clustering”, *IASTED International Conference on Advances in Computer Science and Technology*, pp. 328-333, 2006.

[16]. S. Salvador and P. Chan, “Determining the number of clusters/segments in hierarchical clustering/segmentation algorithms”, *Proc. 16th IEEE Intl. Conf. on Tools with AI*, pp. 576-584, 2004.

[17]. K. Lang, “Learning to filter netnews”, *In Proceedings of the 12th International Conference on Machine Learning*, pp. 331-339, 1995.

[18]. K. McCallum, “Bow: A toolkit for statistical language modeling”, *text retrieval, classification and clustering*. http://www.cs.cmu.edu/#mccallum/bow, 1996.

[19]. W.M. Rand, “Objective criteria for the evaluation of clustering methods”, *Journal of the American Statistical Assoc*, vol. 66, pp. 846-850,1971.